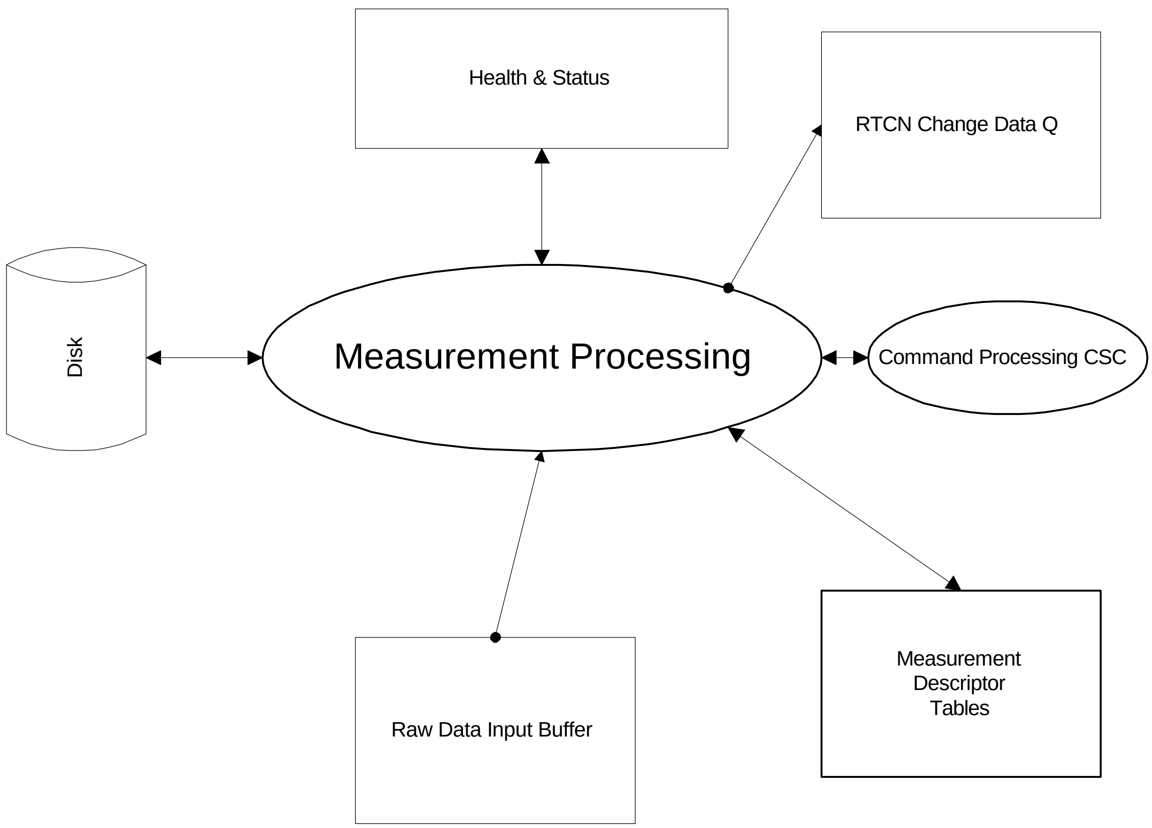
1. PCM DLK MEASUREMENT PROCESSING CSC

1.1 PCM DLK MEASUREMENT PROCESSING CSC Introduction

1.1.1 PCM DLK MEASUREMENT PROCESSING CSC Overview

The PCM Downlink Measurement Processing CSC provides software that will take an input stream of unprocessed data that has been linked to a Measurement Descriptor Table entry and convert the raw data to a processed engineering unit values. The data will then be sent to the RTCN via a Gateway API call.



1.2 PCM DLK MEASUREMENT PROCESSING CSC Specifications

1.2.1 PCM DLK MEASUREMENT PROCESSING CSC Groundrules

- TCID tables must be loaded to the Gateway local media storage prior to initialization.
- The GCP Service API is implemented under the Gateway Common Service CSCI.

1.2.2 PCM DLK MEASUREMENT PROCESSING CSC Functional Requirements

Measurement Processing

1.2.2.1 The PCM Downlink Gateway shall convert measurements to standard IEEE-754 floating-point engineering unit form, if processing parameters for that measurement specify this conversion.

1.2.2.2 The PCM Downlink Gateway shall provide the capability to convert to engineering units as specified by the conversion coefficients for each analog raw value.

1.2.2.3 The PCM Downlink Gateway shall provide the capability to convert 32-bit single precision and 64-bit double precision GPC Floating point data.

1.2.2.4 The PCM Downlink Gateway shall provide the capability to process digital pattern measurements composed of 2 to 16 bits.

1.2.2.5 The PCM Downlink Gateway shall provide the capability to process multi-word digital patterns composed of 2 to 64 bits.

1.2.2.6 The PCM Downlink Gateway shall support the processing of 16-bit parent words which contain multiple measurements by separating the parent word into measurements and processing each measurement individually.

1.2.2.7 The PCM Downlink Gateway shall be capable of processing analog, digital pattern, and discrete group data types included within a parent word.

1.2.2.8 The PCM Downlink Gateway shall maintain analog measurements based on type and/or subtype.

1.2.2.9 The PCM Downlink Gateway shall provide the capability to process some or all bits of a discrete group.

1.2.2.10 *The PCM Downlink Gateway shall provide the capability to process some or all of a discrete group from 1 to 64 bits.(Atlas)*

1.2.2.11 The PCM Downlink Gateway shall process a group of individual data words defined as a Time Homogeneous Data Set (THDS).

1.2.2.12 The PCM Downlink Gateway shall not require words which make up a THDS to be contiguous within the PCM frame.

1.2.2.13 The PCM Downlink Gateway shall allow all data types to be included in a Type II THDS data set.

1.2.2.14 The PCM Downlink Gateway shall provide the capability to isolate and process individual components within parent words.

1.2.2.15 The PCM Downlink Gateway shall process all data types defined in the SLS (82K00200 Appendix B).

Data Change Detection

1.2.2.16 The PCM Downlink Gateway shall maintain the current value of each significantly changed measurement.

1.2.2.17 The PCM Downlink Gateway shall compare each acquired measurement with the previous value of the measurement to detect a change in the measurement.

1.2.2.18 Any change of the acquired measurement from the previous value is a significant change for a digital pattern type measurement.

1.2.2.19 The PCM Downlink Gateway shall perform significant change checks against unprocessed data.

1.2.2.20 The PCM Downlink Gateway shall detect a significant change for each unprocessed measurement whenever the absolute difference between the required measurement and the previously recorded measurement is different.

~~1.2.2.21 The PCM Downlink Gateway shall provide the capability to simultaneously check from one to all bits of a discrete group.~~

1.2.2.21 The PCM Downlink Gateway shall provide the capability to test each discrete measurement to detect any change in data.

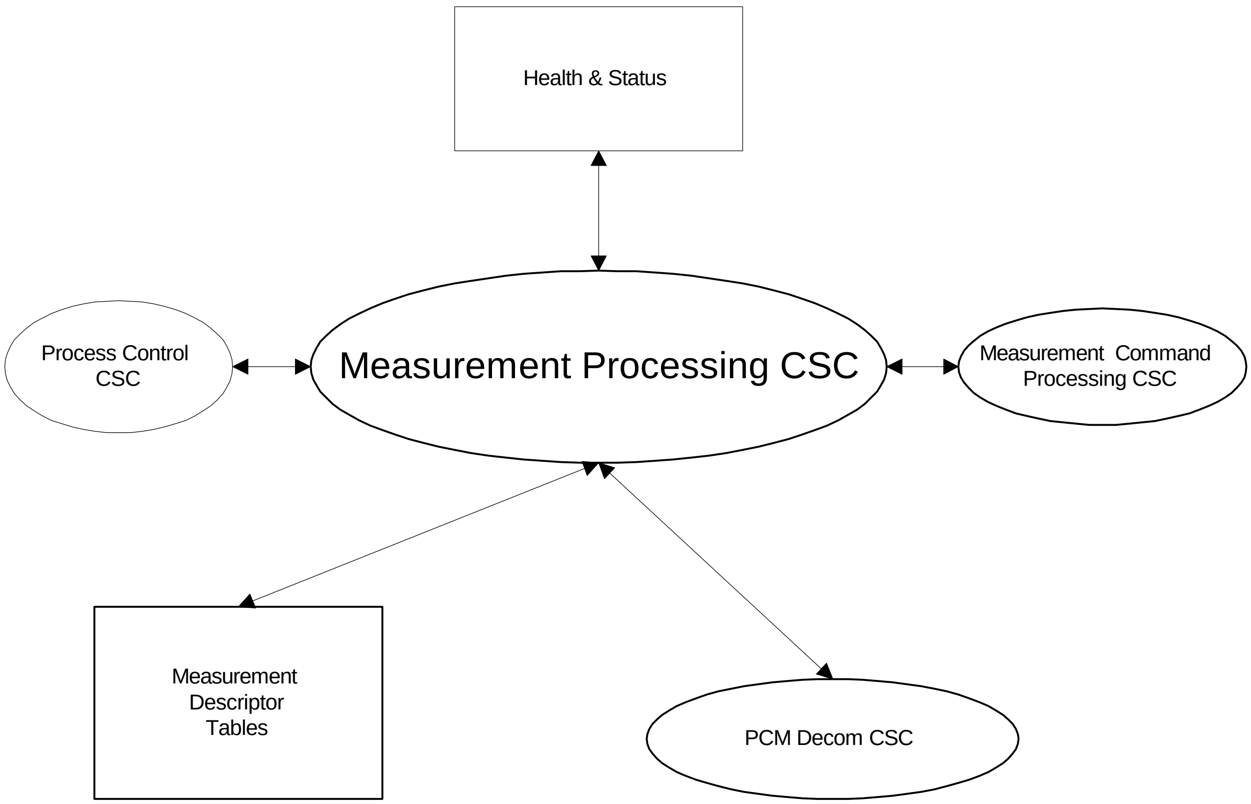
1.2.3 PCM DLK COMMON PROCESSING CSC Performance Requirements

.

The PCM Downlink Gateway shall process 20,000 measurements per second.

1.2.4 PCM DLK COMMAND PROCESSING CSC Interfaces Data Flow Diagrams

External Data Flow Diagram



The Measurement Processing CSC interfaces with a number of processes. The main interface is the incoming raw data stream that contains unprocessed data and a pointer to the MDT from the Telemetry Interface CSC. The Measurement Processing CSC is spawned by the Initialization CSC. Commands to control the processing of measurements will come from the Command Processing CSC. The Measurement Command Processing CSC will also co-ordinate with the Measurement Processing CSC when a change to the tables is made. The Measurement Descriptor Tables are used to determine what the incoming raw data is and how to process it.

1.3 CSC Name Design Specification

Include a brief description of the architecture of the CSC. If there are any priorities associated with the design they should be specified here.

1.3.1 CSC Name Detailed Data Flow

This data flow provides a pictorial representation of the data flow between external sources and destinations and the major and minor functions of the CSC. This is an example detailed data flow diagram.

Detailed Data Flow Diagram Example

The purpose of the Detailed Interface/Data Flow diagram is to show all of the interfaces, internal as well as external, of the CSC. It conveys in a pictorial format all of the input and output streams that the CSC deals with, but not their content. The content of each of these streams of data is described in the detailed design below. Include with the diagram a short paragraph describing the data flow so that a reader can pick up the document and understand the data flow without a conversation with the developer.

Note: Do not use the underscore_character between words. This is an English language document that we want people to be able to pick up and read in their natural language, not a pseudo programming language.

1.3.2 CSC Name External Interfaces

1.3.2.1 CSC Name Message Formats

This data is the System Messages output by the CSC. TBD

1.3.2.2 CSC Name Display Formats

None.

1.3.2.3 CSC Input Formats

None

1.3.2.4 Recorded Data

This area contains a table of data that is recorded by the CSC. All data being recorded must be contained in the table. List the name of each message recorded, the type of recording (listed in the example), and location the data is recorded to. Data will be recorded on the local storage device only when approved by the Design Panel. These approvals can be expected to be rare.

Name of Recorded Data	Recording Type	SDC	Local
		X	
		X	
		X	
		X	
		X	

1.3.2.5 CSC Name Printer Formats

None.

1.3.2.6 Interprocess Communications

TBS

1.3.2.7 CSC External Interface Calls (e.g., API Calling Formats)

TBS

1.3.2.8 CSC Name Table Formats

TBS

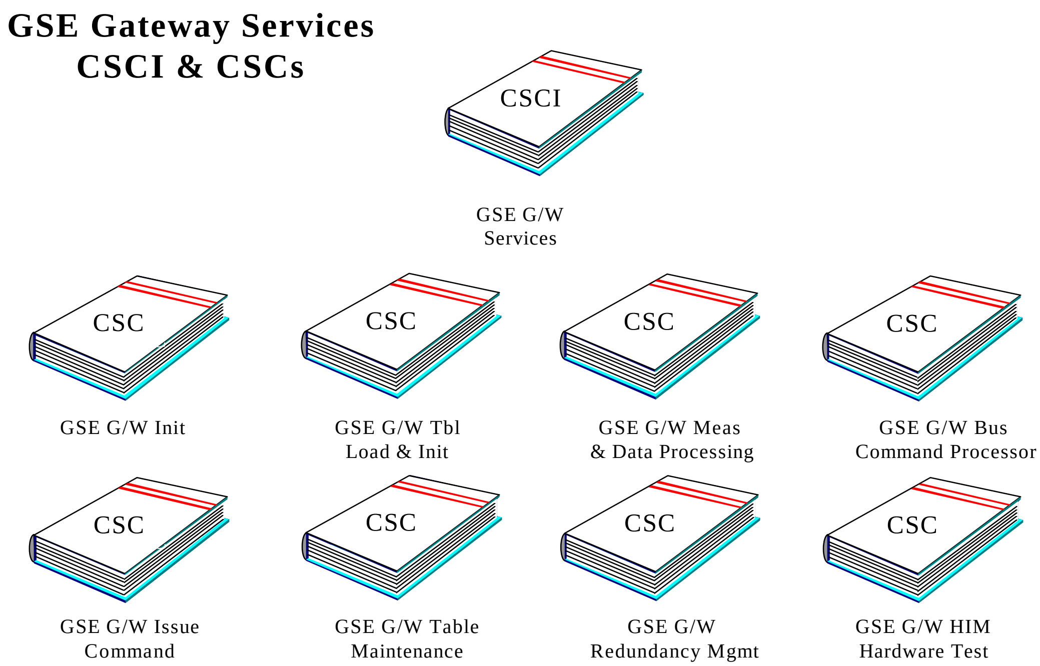
1.3.3 CSC Name Test Plan

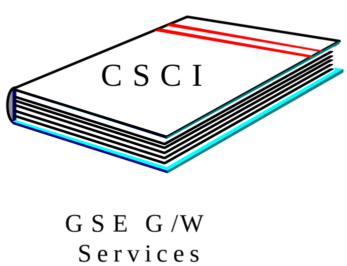
Example:

TBS

Appendix A

In most cases a CSCI is composed of the functionality contained in multiple CSCs that are part of the CSCI. The intent of the SRS/Design Specification is to capture the requirements and design of both the CSCI and its CSCs in one set of volumes. Since we are using electronic forms of documentation for the most part these should be easily captured in one place with links to others? At any rate one option for the structure of a CSCI document is as follows:



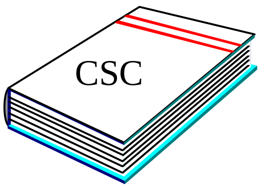


<u>CSCI Outline</u>

1.0 CSCI Name
1.1 CSCI Name Introduction
1.2 CSCI Name Overview
1.2.1 CSC 1 Name Document
1.2.2 CSC 2 Name Document
1.2.3 CSC 3 Name Document
o
o
o
1.2.N CSC N Name Document

Design Panel 2

CSC Requirements Spec

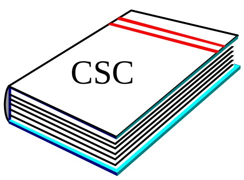


GSE G/W Init

1.0 GSE G/W Init
1.1 GSE G/W Init Introduction
1.1.1 GSE G/W Init Overview
1.1.2 GSE G/W Init Operational Description
1.2 GSE G/W Init Specifications
1.2.1 GSE G/W Init Groundrules
1.2.2 GSE G/W Init Requirements
1.2.3 GSE G/W Init Performance Requirements
1.2.4 GSE G/W Init Interfaces/Data Flow Diagrams

Design Panel 3

CSC Requirements Spec



GSE G/W Init

1.0 GSE G/W Init
1.1 GSE G/W Init Introduction
1.1.1 GSE G/W Init Overview
1.1.2 GSE G/W Init Operational Description
1.2 GSE G/W Init Specifications
1.2.1 GSE G/W Init Groundrules
1.2.2 GSE G/W Init Requirements
1.2.3 GSE G/W Init Performance Requirements
1.2.4 GSE G/W Init Interfaces/Data Flow Diagrams

Present only Changes

CSC Design Spec

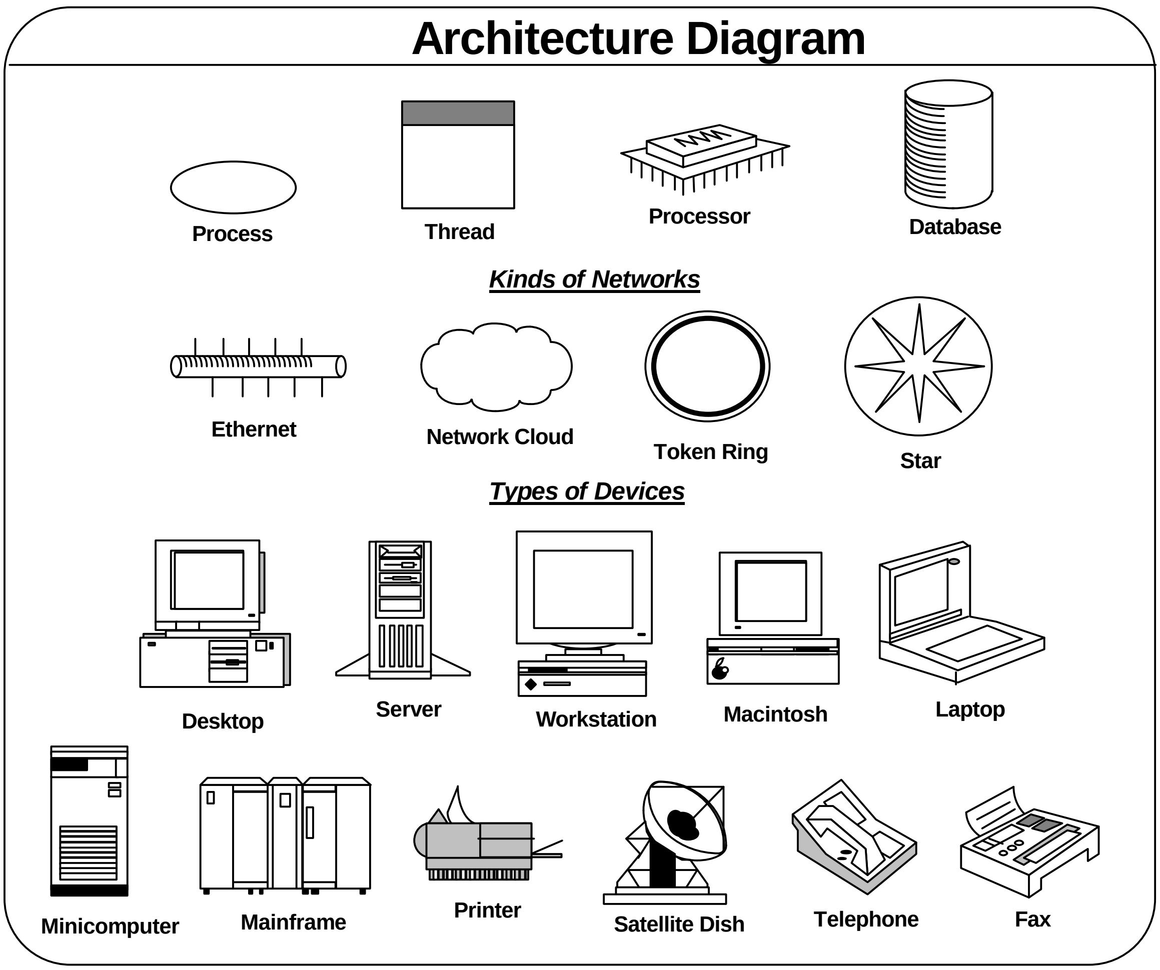
1.3 GSE G/W Init Design Specification
1.3.1 GSE G/W Init Detailed Data Flow
1.3.2 GSE G/W External Interfaces
1.3.3 GSE G/W Internal Interfaces
1.3.4 GSE G/W Structure Diagram
1.3.5 GSE G/W Test Plan

Appendix B

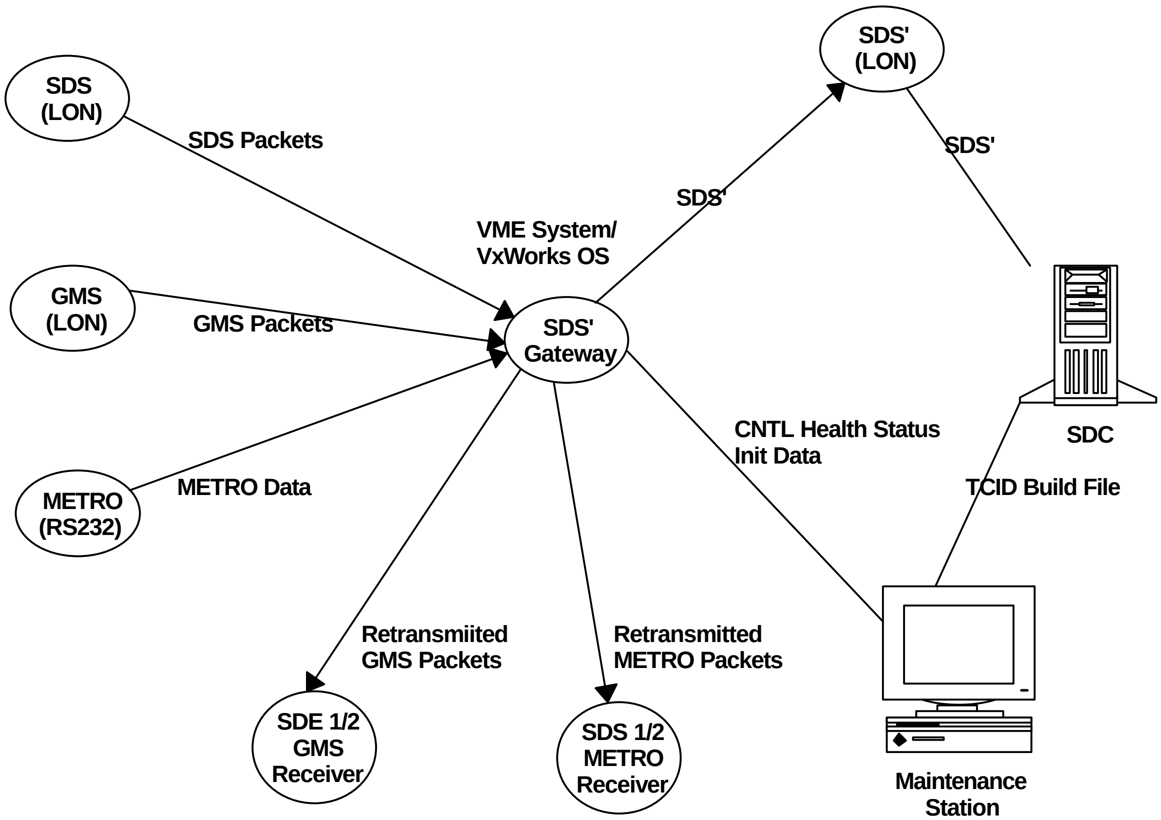
This appendix provides information that will be used for CSCs that are developed using Object Oriented Design (OOD) methodology.

Note: The information in Appendix B is preliminary and will change as the OOD methodology to be used on the CLCS project matures.

The OMT/Rumbaugh notation will be used to document the architecture according to the following legend:



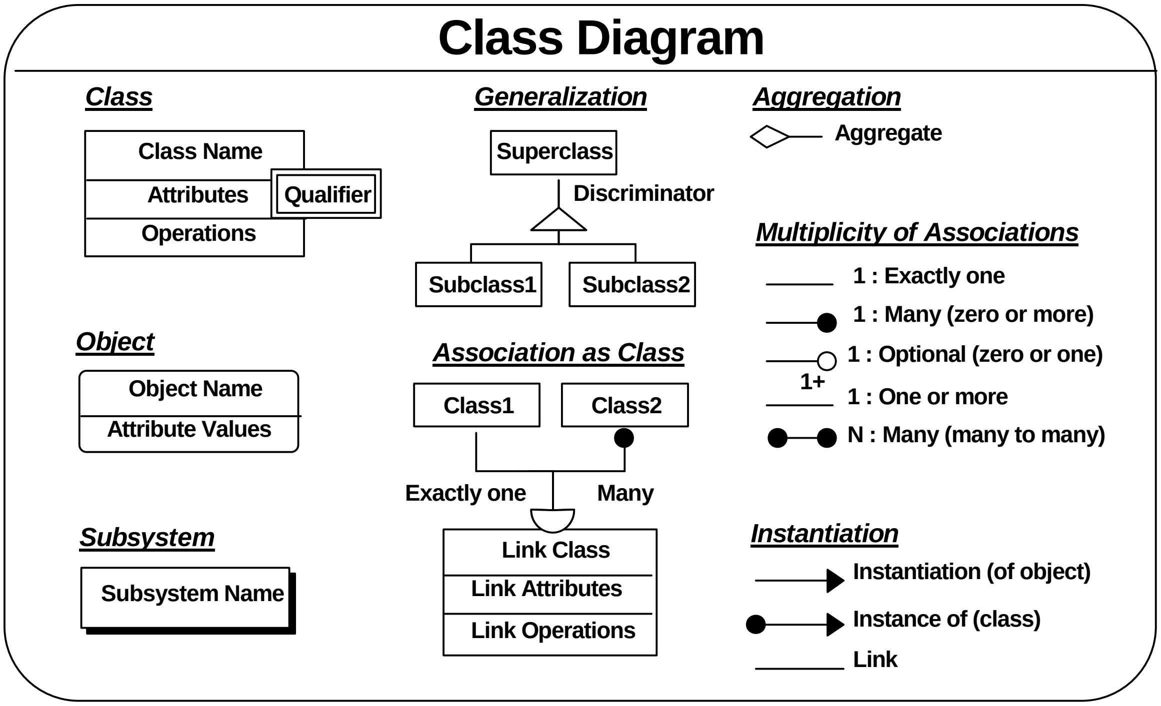
An example architecture diagram is shown below:



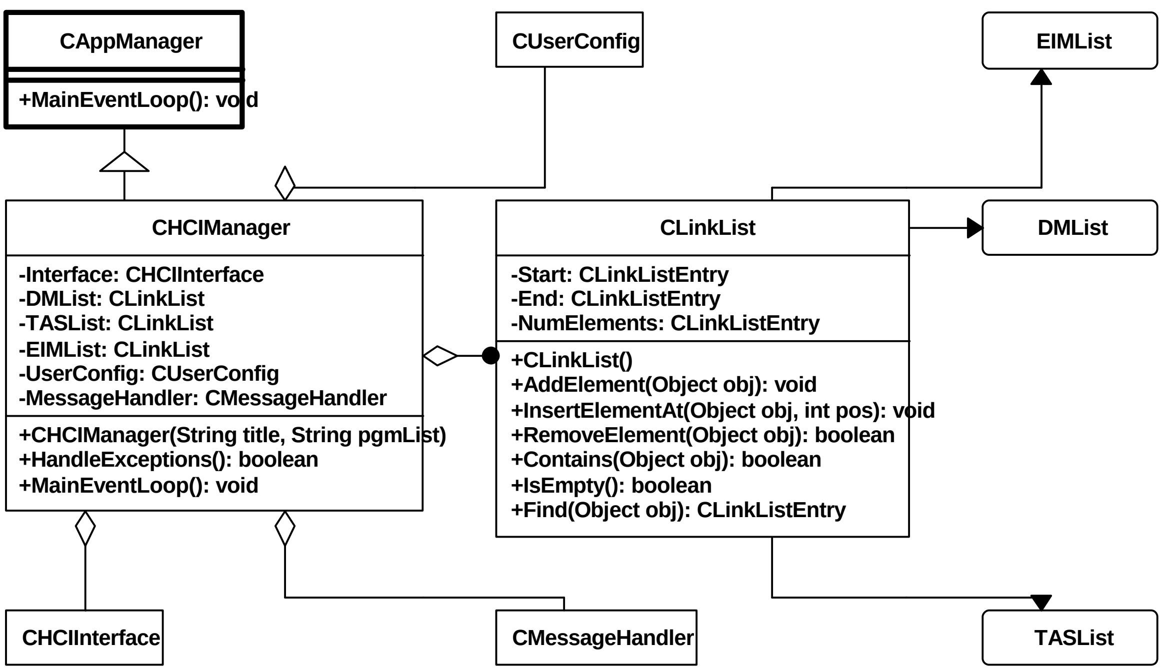
1.4 DP2

Prior to DP2, the designer will decide whether to use Structured Design or OOD. If structured design is to be used, the examples used in the body of this document will be used. If OOD is to be used, the CSC designer should provide a set of preliminary class diagrams. At DP2, the focus is on requirements and preliminary design. Accordingly, the class diagrams presented at DP2 should describe the key abstractions in the requirements (“problem space” classes).

The OMT/Rumbaugh notation will be used to document class information according to the following legend:



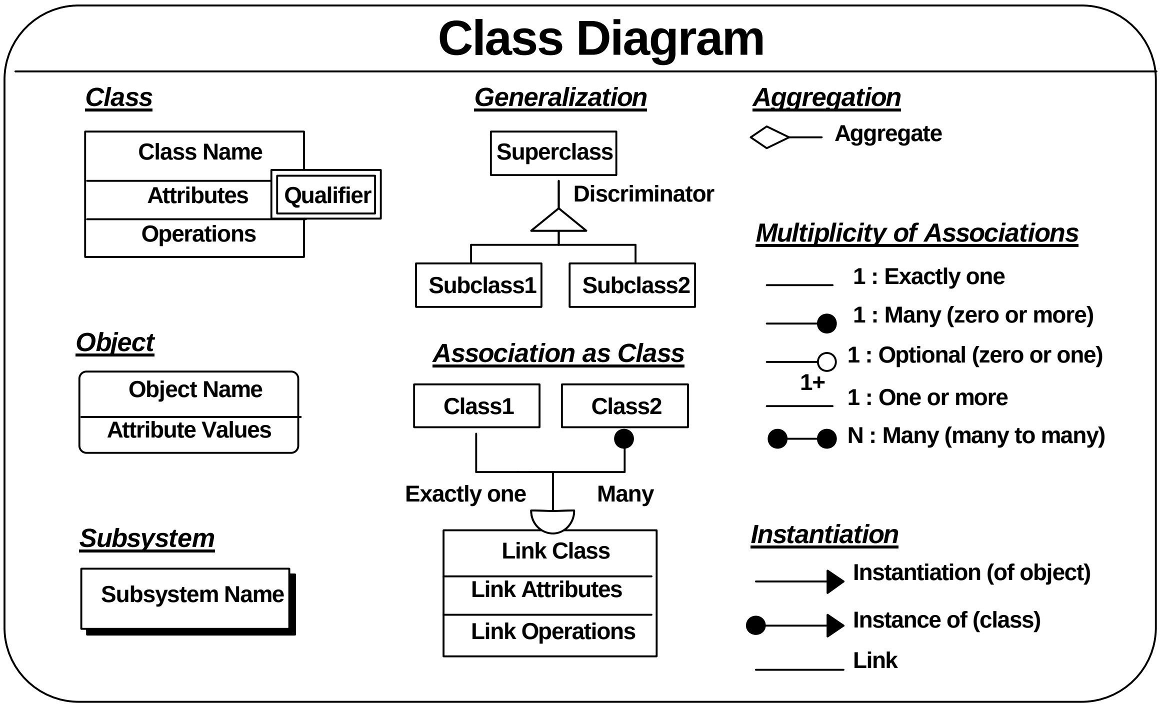
An example class diagram is shown below:



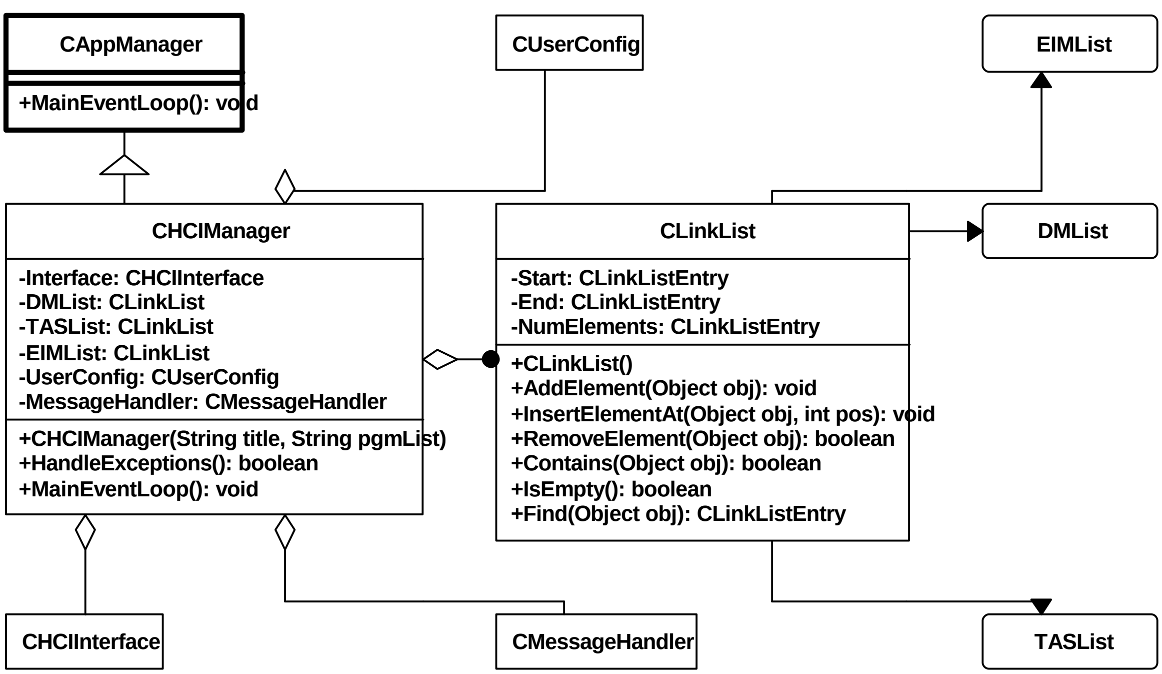
1.5 DP3

For DP3, the designer should provide a set of additional class diagrams that augment the analysis phase classes presented at DP2. Further refinements of the classes presented at DP2 may be included to provide context and aid in understanding. The purpose of these diagrams are to document the mechanisms that will be used to implement the requirements for this CSC.

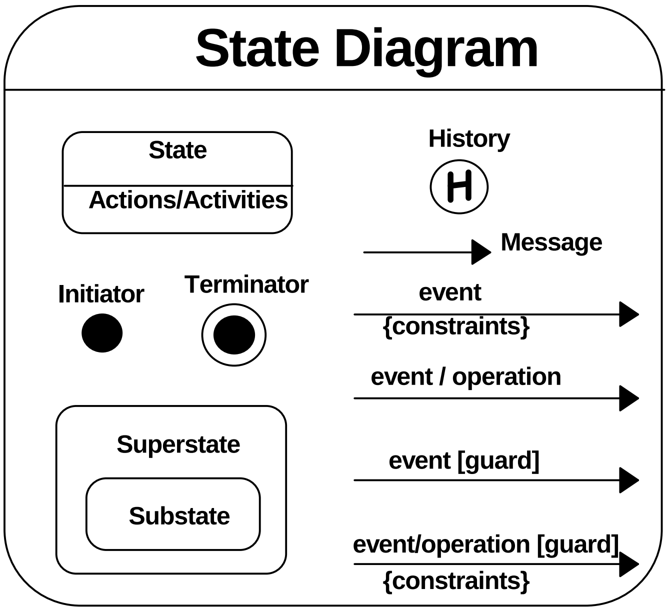
The OMT/Rumbaugh notation will be used to document class information according to the following legend:



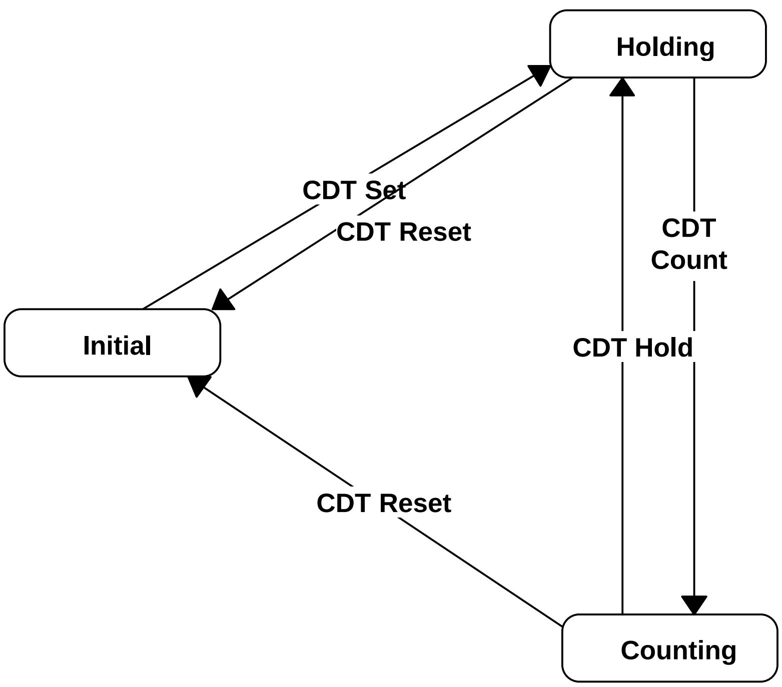
An example class diagram is shown below:



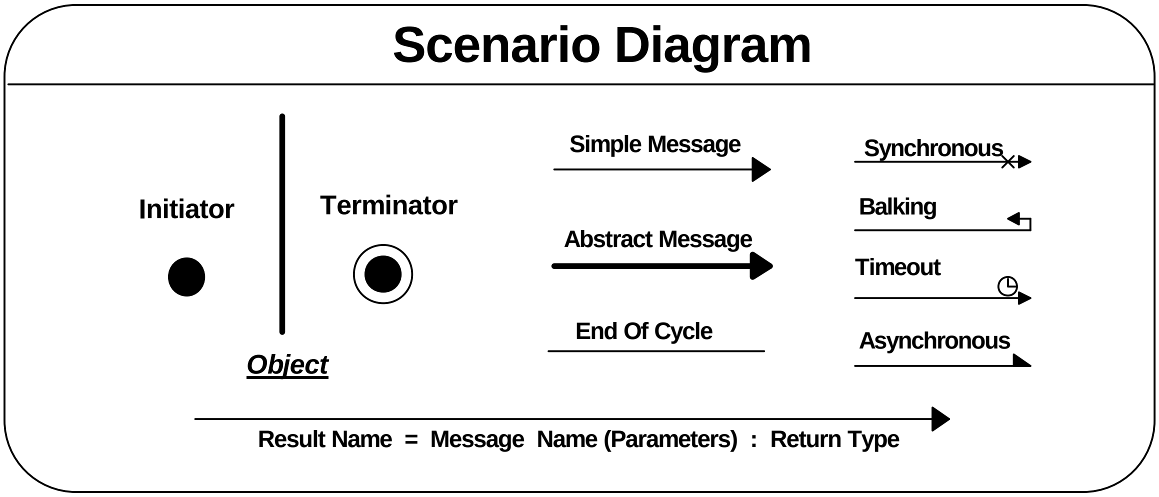
In addition, for classes that exhibit state machine behavior, a state diagram should be included that depicts the dynamic behavior of the class according to the following legend:



An example state diagram for a software implementation of a countdown clock is shown below:



In addition, key scenarios of the CSC's operation or key processing threads should be documented with an scenario diagram according to the legend below:



An example scenario diagram that illustrates the message flow responsible for opening a specific valve is shown below: